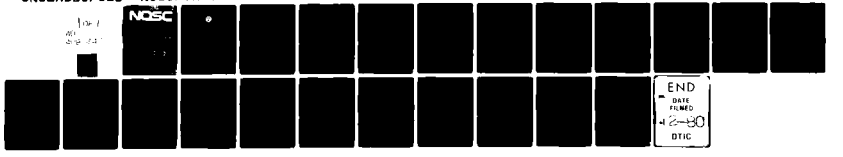
AD-A091 346 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 12/1
A LEAST SQUARES ADAPTIVE LATTICE EQUALIZER ALGORITHM.(U)
SEP 80 E H SATORIUS, J D PACK
UNCLASSIFIED NOSC/TR-575 NL



LEVEL II

12

NOSC

NOSC TR 575

NOSC TR 575

AD A091346

DDC FILE COPY

Technical Report 575

A LEAST SQUARES ADAPTIVE LATTICE EQUALIZER ALGORITHM

E.H. Satorius
J.D. Pack

2 September 1980

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

The work discussed in this report was done under the sponsorship of the Center's Independent Research and Independent Exploratory Development (IRIED) program during the period of October 1978 through September 1979.

Released by
PM Reeves, Acting Head
Electronics Division

Under Authority of
RH Hearn, Head
Fleet Engineering Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NOSC Technical Report 575 (TR 575)

2. GOVT ACCESSION NO.
AD-A091 346

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Least Squares Adaptive Lattice Equalizer Algorithm

5. TYPE OF REPORT & PERIOD COVERED
IRIED - October 1978 Through September 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
E. H. Satorius [redacted] J. D. Pack

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Ocean Systems Center
San Diego, California 92152

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61152N - ZR000/01
ZR014 08 11 632-ZR94

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
2 Sep[redacted]80

13. NUMBER OF PAGES
22

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Equalization
Adaptive
Lattice

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In many applications of adaptive data equalization, rapid initial convergence of the adaptive equalizer is of paramount importance. Apparently the fastest known equalizer adaptation algorithm is based on a least squares estimation algorithm. The least squares algorithm, which is a special case of the Kalman estimation algorithm, was first applied to channel equalization by Godard in a seminal paper. One disadvantage with the Godard algorithm is that the complexity, i.e., number of additions and multiplications, of the algorithm grows quadradically with the number of filter coefficients. Recently, however, Morf, Ljung, Lee and others have shown how the complexity of the conventionally implemented least squares algorithms (e.g., Godard's

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

algorithm) can be made to grow only linearly with the number of filter coefficients. Furthermore, these computationally simpler least squares algorithms may be implemented either in tapped delay line or lattice form. The application of the tapped delay line form, i.e., the fast Kalman algorithm, to channel equalization has been considered recently by Falconer and Ljung. In this paper, it is shown how the least squares lattice algorithms originally introduced by Morf and Lee can be adapted to the equalizer adjustment algorithm. The extremely rapid start up convergence properties of the least squares lattice equalizer are confirmed by computer simulation.

SECTION 1. INTRODUCTION

Traditionally, equalization of a baseband pulse amplitude modulated (PAM) system has been accomplished with a tapped delay line (TDL) filter.[1] However, the application of lattice structures to channel equalization was presented in a recent report by Satorius and Alexander.[2] Lattice structures offer a number of potential advantages over TDL filters. First, an N-stage lattice filter automatically generates all of the outputs which could be generated by N different TDL filters with lengths ranging from 1 to N filter taps. This property of lattice structures allows the dynamic assignment of the particular length of the lattice equalizer that proves most effective at any instant of equalization. A second advantage is that longer lattice filters may be built up from shorter ones simply by adding on more lattice stages. This property should prove useful in designing large scale systems which employ lattice equalizers. Finally, an important property of lattice structures in general is their high insensitivity to round-off noise.[3] The implications of this property with regard to adaptive lattice equalization will not be examined in this paper but that is an important area for future investigation.

This report is concerned primarily with the adaptation algorithm used to update the filter coefficients in the lattice equalizer. In particular it will demonstrate how the recently developed least squares lattice estimation algorithms of Morf, Lee and others[4,5,6,7] can be used to provide the extremely rapid start up performance of the lattice equalizer. This rapid initial convergence is characteristic of least squares algorithms as observed in References 8, 9 and 10. These references note, by contrast, gradient estimation algorithms tend to converge much more slowly during the initial training portion of equalization. This slower convergence behavior is also characteristic of the gradient lattice algorithms (which were examined in Reference 2) as compared with the least squares lattice algorithms. Comparisons between gradient lattice and least squares lattice equalizers will be presented in this paper. First we will briefly review the general least squares problem from which the least squares lattice equalizer arises, then, in Section 3, present the least squares lattice equalizer algorithm and, finally, in Section 4, the simulation results.

SECTION 2. LEAST SQUARES CHANNEL EQUALIZATION

As noted in Reference 10, apparently the fastest converging equalizer algorithm (which is a special case of the Kalman estimation algorithms[8,9,10]) arises in the context of the classical least squares problem: at each sampling instant, find the set of equalizer tap coefficients that minimize the accumulation of the squared errors between the filter output and a desired output up to that time. In particular, suppose a(0), a(1), . . . a(t) is a training sequence of data symbols which are known by the equalizer adjustment algorithm. Further, assume that the a(n) sequence is transmitted over a channel, resulting in a sequence of equalizer inputs, x(0), x(1), . . ., x(n) and define $X_N(n)$ to be an (N+1) - dimensional vector consisting of time-delayed samples of x(n), i.e.,

$$X_N(n) = (x(n), x(n-1), \ldots, x(n-N)^T \quad (1)$$

where superscript T denotes transpose and x(n-i) = 0 when i > n. The least squares problem is equivalent to finding the (N+1) - dimensional vector of equalizer tap values, $F_N(n)$, which minimizes the following sum of squared errors:

$$\sum_{p=0}^{n} w^{n-p} \left| a(p) + F_N^T(n)\, X_N(p) \right|^2 .$$

The parameter w is a real constant, $0<w\leq$, and is included in the sum of squared errors to allow the equalizer to track slow time variations in the channel. Typically, w is close to 1. The inverse of (1-w) is, approximately, the memory of the equalizer.

Differentiating the above sum with respect to the components of $F_N(n)$ and setting the result to zero leads to the following equation for the $F_N(n)$ vector:

$$\mathcal{R}_N(n)\, F_N(n) = - \sum_{p=0}^{n} w^{n-p}\, a(p)\, X_N(p) . \tag{2}$$

In Equation 2, the (N+1) by (N+1) matrix, $\mathcal{R}_N(n)$, is given by:

$$\mathcal{R}_N(n) = \sum_{p=0}^{n} w^{n-p}\, X_N(p)\, X_N^T(p) . \tag{3}$$

The solutions of Equations 2 and 3 provide the least squares equalizer tap vector, $F_N(n)$, at the n^{th} data sample. Notice in the above equations the limits on all the summation signs extend from p=0 to p=n. The lower limit, therefore, imposes the assumption on the data that x(p) = 0 for p = -1, . . . , -N. These limits lead to the so-called "pre-windowed" least squares algorithm (which will be considered in this paper). If the limits are p = N and p = n, the un-windowed or "covariance" algorithm is obtained. A more complete discussion of the different windowing methods may be found in Reference 5.

At this point, it should be noted that solving Equations 2 and 3 could proceed along one of two possible lines of development. In the first $F_N(n)$ obeys the following time update relation (References 8 and 10):

$$F_N(n) = F_N(n-1) - \beta_N(n)\, \xi(n), \tag{4a}$$

where

$$\beta_N(n) = \mathcal{R}_N(n)^{-1}\, X_N(n) , \tag{4b}$$

and

$$\xi(n) = a(n) + F_N(n-1)^T\, X_N(n) . \tag{4c}$$

The major computational burden in this formulation lies in computing the (N+1) - dimensional vector, $\beta_N(n)$, at each iteration. Conventional methods of computing $\beta_N(n)$ (i.e., the Godard algorithm, Reference 8) require on the order of N^2 operations (additions and multiplications) per update. However, by exploiting certain shift properties of $X_N(n)$, it is possible to compute $\beta_N(n)$ at each iteration using a number of operations which depends only linearly on N (i.e., the fast Kalman algorithm, Reference 10). In any case, this formulation (i.e., Equation 4) of the solution to Equation 2 leads to a TDL equalizer structure for which $-F_N(n)$ represents the vector of equalizer taps and $-F_N(n-1)^1\, X_N(n)$ represents the equalizer output.

A second solution to Equation 2 incorporates the lattice method. In this method, certain properties of $\mathcal{R}_N(n)$ (these are discussed in the Appendix) are exploited to derive order update equations for the outputs $y_m(n) = -F_m(n-1)^T X_m(n)$ $(m = 0,1, \ldots, N)$. In particular, lattice recursion relations are derived which relate $y_m(n)$ to $y_{m-1}(n)$ as well as certain other auxiliary variables. In this way all of the least squares outputs, $y_m(n)$ $(m = 0, \ldots, N)$, are automatically generated. This is in contrast to the TDL formulation where only $y_N(n)$ is computed every iteration. The least squares lattice equalizer algorithm is presented in the next section.

SECTION 3. THE LEAST SQUARES LATTICE ALGORITHM

This algorithm is a modified version of the least squares algorithm originally presented in References 4, 5, 6, and 7. The Appendix contains a derivation of the algorithm. A more complete discussion of least squares lattice algorithms in general and their properties may be found in References 4 through 7 and 11 through 13.

In the least squares adaptive lattice equalizer (LSALE) algorithm, the following scalar quantities are stored and updated every iteration:

(1) $\epsilon_m^e(n), \epsilon_m^r(n), r_m(n), e_m(n), \bar{e}_m(n), \bar{k}_m(n), y_m(n)$ $(m = 0,1, \ldots, N)$.

(2) $\gamma_m(n), k_m(n)$ $(m = 0,1, \ldots, N-1)$.

To initialize the algorithm, we set:

$$\epsilon_m^e(-1) = \epsilon_m^r(-2) = \epsilon_m^r(-1) = \delta \tag{5a}$$

$$r_m(-1) = k_m(-1) = \bar{k}_m(-1) = \gamma_m(-1) = 0. \tag{5b}$$

Then, the following computations are performed starting with n = 0:

$$e_o(n) = r_o(n) = x(n) \tag{6a}$$

$$\epsilon_o^e(n) = \epsilon_o^r(n) = w\ \epsilon_o^e(n-1) + x^2(n) \tag{6b}$$

$$y_{-1}(n) = \gamma_{-1}(n) = \gamma_{-1}(n-1) = 0 \tag{6c}$$

$$\bar{e}_{-1}(n) = a(n)\ , \tag{6d}$$

$$k_m(n) = w\, k_m(n-1) + (1-\gamma_{m-1}(n-1))\, e_m(n)\, r_m(n-1) \tag{7a}$$

$$e_{m+1}(n) = e_m(n) - \frac{k_m(n-1)\, r_m(n-1)}{\epsilon_m^r(n-2)} \tag{7b}$$

$$r_{m+1}(n) = r_m(n-1) - \frac{k_m(n-1)}{\epsilon_m^e(n-1)}\, e_m(n) \tag{7c}$$

$$\epsilon^{e}_{m+1}(n) = \epsilon^{e}_{m}(n) - \frac{k^{2}_{m}(n)}{\epsilon^{r}_{m}(n-1)} \tag{7d}$$

$$\epsilon^{r}_{m+1}(n) = \epsilon^{r}_{m}(n-1) - \frac{k^{2}_{m}(n)}{\epsilon^{e}_{m}(n)} \tag{7e}$$

$$\gamma_{m}(n) = \gamma_{m-1}(n) + \frac{\left\{(1-\gamma_{m-1}(n))\, r_{m}(n)\right\}^{2}}{\epsilon^{r}_{m}(n)} \tag{7f}$$

$$y_{m}(n) = y_{m-1}(n) + \frac{\bar{k}_{m}(n-1)}{\epsilon^{r}_{m}(n-1)} r_{m}(n) \tag{7g}$$

$$\bar{e}_{m}(n) = a(n) - y_{m}(n) \ . \tag{7h}$$

The recursions in Equations 7a through 7f are computed for m = 0,1, . . . ,N-1. The recursions in Equations 7g and 7h are computed for m = 0,1, . . . ,N. Finally, the coefficients, $\bar{k}_{m}(n)$, are updated by the recursion:

$$\bar{k}_{m}(n) = w\,\bar{k}_{m}(n-1) + (1-\gamma_{m-1}(n))\ \bar{e}_{m-1}(n)\ r_{m}(n) \tag{8}$$

$$m = 0, \ldots, N.$$

It should be noted that δ in Equation 5a is a small positive constant which is used to ensure nonsingularity of the $\mathcal{R}_{N}(n)$ matrix.

Equations 5 through 8 represent the LSALE algorithm. The scalar, $y_{N}(n)$, represents the LSALE output. A schematic diagram of the lattice equalizer is presented in Figures 1a and 1b. During the initial phase of operation the training sequence, [a(0), a(1), . . . , a(t)], is known to the equalizer. Using these data, the equalizer determines the channel characteristics and adapts to them. As noted in Reference 2, when actual data are being transmitted, the LSALE output, $y_{N}(n)$, is first thresholded to produce a reference sequence, $\hat{a}(n)$. Then $\bar{e}_{m}(n)$ is computed as in Equation 7h with a(n) replaced by $\hat{a}(n)$, and $\bar{k}_{m}(n)$ is updated by Equation 8. It should be noted that the above formulation of the LSALE is for the scalar input case (i.e., a linear equalizer). Extensions to the vector input case, which is required, for instance, in the case of decision feedback equalization (Reference 10), are straightforward and are discussed further in References 4 through 7.

A count of the number of operations needed to calculate each equalizer output and update the LSALE algorithm, Equations 5 through 8, indicates that approximately 19L-12 multiplications and 11L-7 additions are required where L=N+1 is the total number of equalizer taps. This counts divisions as multiplications. It is interesting to compare these figures with those corresponding to the fast Kalman[10] as well as the gradient lattice equalizer algorithm, referred to as ALCE in Reference 2. The results are summarized as:

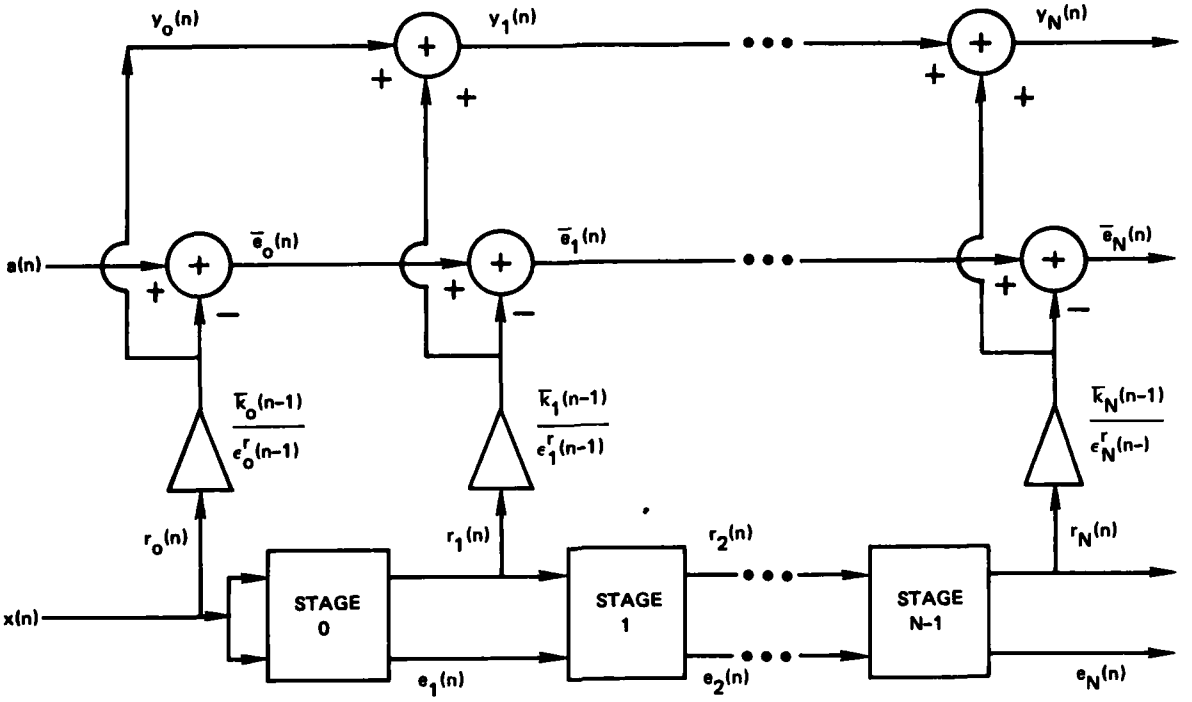


Figure 1a. The least squares, adaptive lattice equalizer.

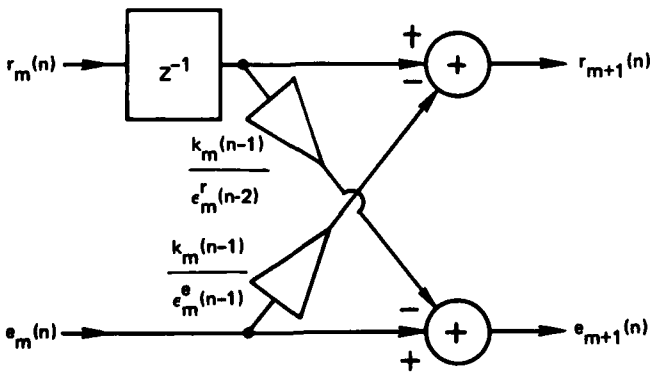


Figure 1b. The mth stage of the lattice.

Algorithm	Multiplications	Additions
LSALE	19L-12	11L-7
ALCE	13L-8	10L-7
Fast Kalman	10L+4	12L+5

For all of these algorithms, divisions are counted as multiplications. As is seen, both of the lattice algorithms have more multiplications than the TDL fast Kalman algorithm. This is due to the power normalizations that are required at each stage of the lattice. It is interesting to note that LSALE is more complex than ALCE. This increased complexity is the price which must be paid for the faster convergence rate of the LSALE algorithm. The improved convergence performance of LSALE over ALCE will be observed in the next section.

SECTION 4. SIMULATION RESULTS

For purposes of examining the convergence performance of LSALE, we have simulated this algorithm for two channels representing pure, heavy amplitude distortion. The impulse response structures of these channels are identical to those considered in Reference 2. The two channels had eigenvalue disparity ratios, that is, ratio of largest-to-smallest eigenvalues of channel correlation matrix, of 11 and 21. In all simulations 11-tap equalizers were used, N = 10 in Equations 7 and 8. The symbol sequence a(n) was a known, random bipolar training sequence (a(n) = ±1), suitably delayed so that the minimum mean square error TDL equalizer taps would be symmetric about the center of the equalizer, as discussed in Reference 2. Also, a small amount of uncorrelated Gaussian noise, noise variance = .001, was added to the output of the channel.

The results of the simulations are presented in Figures 2 and 3, corresponding to eigenvalue ratios of 11 and 21, respectively. In these figures, δ in Equation 5a was set equal to .001 and w in Equations 6 through 8 was set equal to 1. Additional simulation revealed that the start up performance of LSALE was highly insensitive to the choice of δ. It was found in our simulation studies that the best start up performance for the LSALE algorithm was obtained by suppressing the update of the $\bar{k}_m(n)$ coefficients in Equation 8, i.e., holding $\bar{k}_m(n) = 0$, m = 0, . . . ,N, until after the main part of the first data pulse was received. The number of iterations in Figures 2 and 3, however, is reckoned from the initial updating of Equations 6 and 7 and not from the initial updating of the $\bar{k}_m(n)$ coefficients. The improved performance of LSALE which results from suppressing the $\bar{k}_m(n)$ update is analogous to the improved start up performance of the fast Kalman equalizer which results from initially suppressing the tap update, $F_N(n)$ in Equation 4a, as discussed in Reference 10.

For sake of comparison, Figures 2 and 3 include corresponding learning curves for the gradient lattice algorithm (ALCE) described in Reference 2, as well as the simple gradient TDL equalizer algorithm, also described in Reference 2. In fact, the parameters of the ALCE algorithm and the simple TDL gradient algorithm are identical to those parameters used in Reference 2 which provided the best start up performance, i.e., the P_o = .075 curves in Figures 5 and 6 of that reference. As is clearly seen in Figures 2 and 3, the convergence rate of both lattice algorithms is highly insensitive to the channel eigenvalue disparity. However, the LSALE algorithm converges in approximately 40 iterations for both channels,

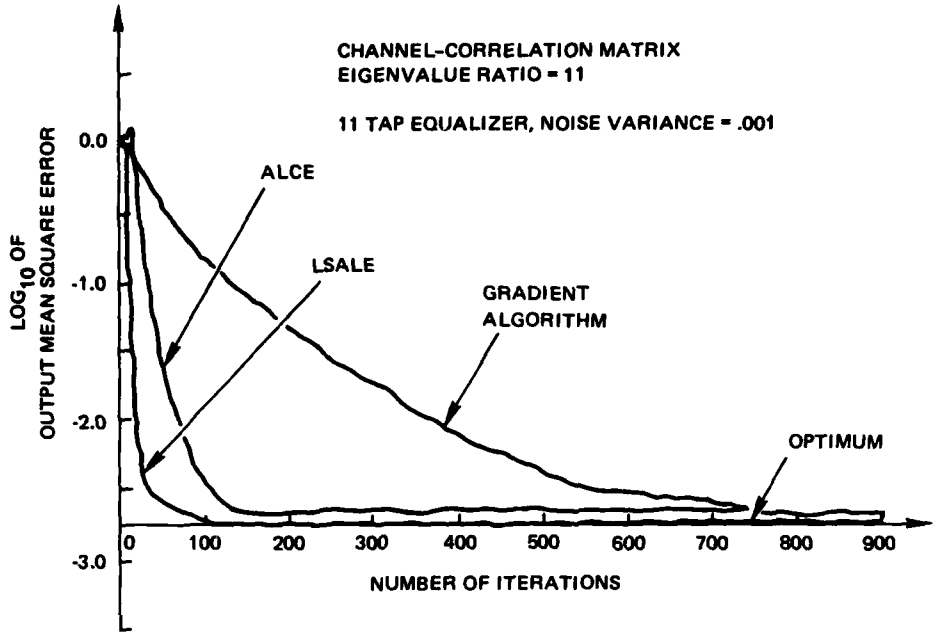


Figure 2. Comparison by simulation of convergence properties for eigenvalue ratio = 11.

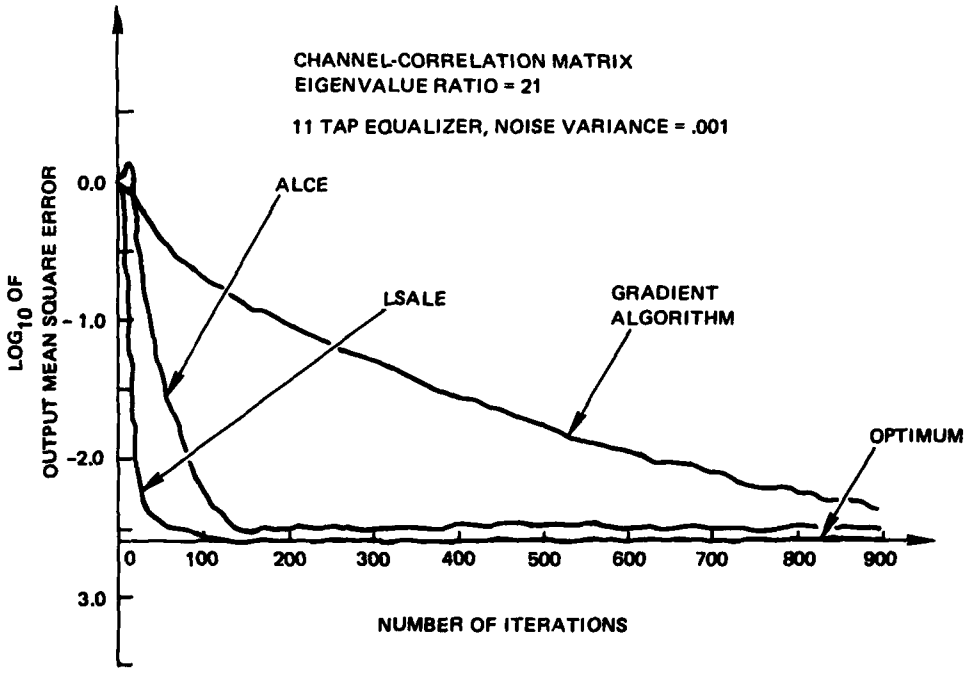


Figure 3. Comparison by simulation of convergence properties for eigenvalue ratio = 21.

whereas ALCE required approximately 120 iterations to converge. The improved convergence performance of both lattice algorithms over the TDL gradient algorithm is clearly observed.

SECTION 5. CONCLUSIONS

In this report we have shown how the new least squares, lattice algorithms may be applied to data equalization. In particular, a new least squares adaptive lattice equalizer (LSALE) algorithm was presented and the improved initial convergence performance of LSALE over gradient based algorithms (implemented in either TDL or lattice structure) was clearly observed in Section 4. Additional properties of least squares lattice algorithms are examined in References 12 and 13.

APPENDIX: DERIVATION OF THE LEAST SQUARES ADAPTIVE LATTICE EQUALIZER ALGORITHM

In deriving the least squares lattice algorithm, considerable use is made of the following properties of the $\mathcal{R}_m(n)$ matrix, which may easily be derived from its definition, Equation 3:

$$\mathcal{R}_m(n) = \left[\begin{array}{c|c} q_m(n) & Q_m^T(n) \\ \hline Q_m(n) & \mathcal{R}_{m-1}(n-1) \end{array}\right] \tag{A1a}$$

$$= \left[\begin{array}{c|c} \mathcal{R}_{m-1}(n) & V_m(n) \\ \hline V_m^T(n) & v_m(n) \end{array}\right] \tag{A1b}$$

The dashed lines in Equations A1a and A1b are used to denote partitioning. The quantities $q_m(n)$ and $v_m(n)$ denote scalars which are given by:

$$q_m(n) = \sum_{p=0}^{n} w^{n-p}\, x^2(p) \ , \tag{A2a}$$

and

$$v_m(n) = \sum_{p=0}^{n} w^{n-p}\, x^2(p-m) \ . \tag{A2b}$$

Also, $Q_m(n)$ and $V_m(n)$ in Equations A1a and A1b denote m-dimensional vectors given by:

$$Q_m(n) = \sum_{p=0}^{n} w^{n-p}\, x(p)\, X_{m-1}(p-1) \ , \tag{A2c}$$

and

$$V_m(n) = \sum_{p=0}^{n} w^{n-p}\, x(p-m)\, X_{m-1}(p) \ . \tag{A2d}$$

A third property of the $\mathcal{R}_m(n)$ matrix is the following time shift relation:

$$\mathcal{R}_m(n) = w\, \mathcal{R}_m(n-1) + X_m(n)\, X_m^T(n) \ . \tag{A3}$$

As in the case of the fast Kalman algorithm,[10] the least squares lattice equalizer is intimately related to the one-step, backward and forward least squares predictor. Therefore, the following sums of squared errors are introduced:

$$\sum_{p=0}^{n} w^{n-p} \; e_m^2(p,n) \; ,$$

and

$$\sum_{p=0}^{n} w^{n-p} \; r_m^2(p,n) \, ,$$

where

$$e_m(p,n) = x(p) + A_m^T(n) \; X_{m-1}(p-1) \; , \tag{A4a}$$

and

$$r_m(p,n) = x(p-m) + B_m^T(n) \; X_{m-1}(p). \tag{A4b}$$

The quantities $e_m(p,n)$ and $r_m(p,n)$ are called the m^{th} order forward and backward prediction error residuals, respectively, and the m-dimensional vectors $A_m(n)$ and $B_m(n)$ in Equation A4 are chosen to minimize the above exponentially weighted sums of squared errors. Differentiating these sums with respect to the coefficients of the $A_m(n)$ and $B_m(n)$ vectors and setting the results equal to zero leads to the following augmented matrix equations:

$$\mathcal{R}_m(n) \, \bar{A}_m(n) = \begin{bmatrix} \epsilon_m^e(n) \\ \\ 0_m \end{bmatrix} \tag{A5a}$$

and

$$\mathcal{R}_m(n) \, \bar{B}_m(n) = \begin{bmatrix} 0_m \\ \\ \epsilon_m^r(n) \end{bmatrix} \; . \tag{A5b}$$

In Equation A5 0_m denotes the m-dimensional zero vector. Also, $\bar{A}_m(n)$ and $\bar{B}_m(n)$ are the following extended (m+1) - dimensional vectors:

$$\bar{A}_m(n) = \begin{bmatrix} 1 \\ \\ A_m(n) \end{bmatrix} , \tag{A6a}$$

and

$$\bar{B}_m(n) = \begin{bmatrix} B_m(n) \\ \\ 1 \end{bmatrix} . \tag{A6b}$$

Finally, in Equation A5, $\epsilon_m^e(n)$ and $\epsilon_m^r(n)$ are the minimum of the above exponentially weighted sums of squared errors:

$$\epsilon_m^e(n) = \min \left\{ \sum_{p=0}^{n} w^{n-p} \; e_m^2(p,n) \right\}$$

$$= \sum_{p=0}^{n} w^{n-p} \; x^2(p) + A_m^T(n) \, Q_m(n) \; , \tag{A7a}$$

and

$$\epsilon_m^r(n) = \min \left\{ \sum_{p=0}^{n} w^{n-p} \; r_m^2(p,n) \right\}$$

$$= \sum_{p=0}^{n} w^{n-p} \, x^2(p-m) + B_m^T(n) \; V_m(n) \; . \tag{A7b}$$

As will be seen, the vectors $\bar{A}_m(n)$ and $\bar{B}_m(n)$ play a central role in the development of the least squares, lattice algorithm.

Another auxiliary vector which is crucial to the development of this algorithm is the (m+1) - dimensional vector, $C_m(n)$, which is given by the solution to:

$$R_m(n) \, C_m(n) = X_m(n) \; . \tag{A8}$$

At this point we will concentrate on obtaining order updates for the three vectors, $\bar{A}_m(n)$, $\bar{B}_m(n)$, and $C_m(n)$. In particular,

$$\bar{A}_{m+1}(n) = \begin{bmatrix} \bar{A}_m(n) \\ \\ 0 \end{bmatrix} - \frac{k_m(n)}{\epsilon_m^r(n-1)} \begin{bmatrix} 0 \\ \\ \bar{B}_m(n-1) \end{bmatrix} , \tag{A9}$$

where

$$k_m(n) = V_{m+1}^T(n) \, \bar{A}_m(n) \; . \tag{A10}$$

Equation A9 may be verified by premultiplying its right hand side by $\mathcal{R}_{m+1}(n)$ and noting with the aid of Equations A1 and A5 that the last m+1 elements of the resulting column vector vanish. This implies that the last m+1 elements of the column vector defined by the right hand side of Equation A9 satisfy the same equation that is satisfied by the m+1 elements of $A_{m+1}(n)$ (obtained from Equation A5a with m replaced by m+1). Therefore, assuming that the matrices, $\mathcal{R}_m(n)$, are nonsingular* the last m+1 elements of the column vector defined by the right hand side of Equation A9 must equal the m+1 elements of $A_{m+1}(n)$. Also, the first element of the column vector defined by the right hand side of Equation A9 equals unity. Thus, Equation 9 is verified.

An order recursion for $\epsilon_m^e(n)$ also can be derived by premultiplying the right hand side of Equation A9 by $\mathcal{R}_{m+1}(n)$ and noting that the first element of the resulting column vector must equal $\epsilon_{m+1}^e(n)$. The resulting recursion for $\epsilon_m^e(n)$ is given by

$$\epsilon_{m+1}^e(n) = \epsilon_m^e(n) - \frac{k_m(n)\hat{k}_m(n)}{\epsilon_m^r(n-1)} , \tag{A11}$$

where

$$\hat{k}_m(n) = Q_{m+1}^T(n)\,\bar{B}_m(n-1) . \tag{A12}$$

In the order recursion for $\epsilon_{m+1}^e(n)$, two scalars, $k_m(n)$ and $\hat{k}_m(n)$, appear. These can be shown to be equal by considering the product

$$\phi = (0, \bar{B}_m^T(n-1))\ \mathcal{R}_{m+1}(n)\ (\bar{A}_m^T(n), 0)^T. \tag{A13}$$

From Equations A1b, A5a and A10

$$\phi = (0, \bar{B}_m^T(n-1))\ (\epsilon_m^e(n), 0_m^T, k_m(n))^T = k_m(n). \tag{A14a}$$

On the other hand, using Equations A1a, A5b and A12, we have:

$$\phi = (\hat{k}_m(n), 0_m^T, \epsilon_m^r(n-1))\ (\bar{A}_m^T(n), 0)^T = \hat{k}_m(n) . \tag{A14b}$$

Comparing Equations A14a and A14b shows that $k_m(n) = \hat{k}_m(n)$. Therefore, Equation A11 may be rewritten as

$$\epsilon_{m+1}^e(n) = \epsilon_m^e(n) - \frac{k_m^2(n)}{\epsilon_m^r(n-1)} . \tag{A15}$$

In a manner analogous to the development of Equations A9 and A15, we also can derive the following order update recursions for $\bar{B}_m(n)$ and $\epsilon_m^r(n)$:

---

*In practice, the positive definiteness (and hence nonsingularity) of $\mathcal{R}_m(n)$ can be guaranteed by initializing $\mathcal{R}_m(n)$ to a positive scalar times the identity matrix.

$$\bar{B}_{m+1}(n) = \begin{bmatrix} 0 \\ \bar{B}_m(n-1) \end{bmatrix} - \frac{k_m(n)}{\epsilon_m^e(n)} \begin{bmatrix} \bar{A}_m(n) \\ 0 \end{bmatrix} \tag{A16}$$

and

$$\epsilon_{m+1}^r(n) = \epsilon_m^r(n-1) - \frac{k_m^2(n)}{\epsilon_m^e(n)} . \tag{A17}$$

To complete the development of the order updates, $C_m(n)$ obeys the order recursion:

$$C_m(n) = \begin{bmatrix} C_{m-1}(n) \\ 0 \end{bmatrix} + \frac{r_m(n,n)}{\epsilon_m^r(n)} \bar{B}_m(n) . \tag{A18}$$

Equation A18 may be verified by premultiplying its right hand side by $\mathcal{R}_m(n)$ and noting that from Equations A1b and A8:

$$\mathcal{R}_m(n) \; (C_{m-1}^T(n), 0)^T = (X_{m-1}^T(n), \alpha)^T , \tag{A19}$$

where the constant α is given by (from Equations A1b and A8)

$$\alpha = V_m^T(n) \, C_{m-1}(n) = V_m^T(n) \, \mathcal{R}_{m-1}^{-1}(n) \; X_{m-1}(n) . \tag{A20}$$

However, from Equations A1b, A4b and A5b

$$\alpha = x(n-m) - r_m(n,n) . \tag{A21}$$

Using Equation A5b as well as A19 and A21, it is seen that the product of $\mathcal{R}_m(n)$ with the right hand side of Equation A18 is simply the vector, $X_m(n)$. Thus, the right hand side of A18 satisfies the equation for $C_m(n)$, A8. This verifies Equation A18.

Aside from the order update recursions derived above, we also will need certain time update recursions. In particular,

$$\bar{A}_m(n) = \bar{A}_m(n-1) - \frac{e_m(n,n)}{1 - \gamma_{m-1}(n-1)} \begin{bmatrix} 0 \\ C_{m-1}(n-1) \end{bmatrix} , \tag{A22}$$

where the scalar $\gamma_m(n)$ is given by:

$$\gamma_m(n) = X_m^T(n) \, C_m(n) = X_m^T(n) \, \mathcal{R}_m^{-1}(n) \, X_m(n) . \tag{A23}$$

As can be seen from Equation 23 (see References 5 and 11), $\gamma_m(n)$ is bounded between 0 and 1. Equation A22 may be derived by noting from Equations A1a and A5a that

$$A_m(n-1) = - \mathcal{R}_{m-1}^{-1}(n-2) \, Q_m(n-1). \tag{A24a}$$

Also, from Equation A3

$$Q_m(n) = w\, Q_m(n-1) + x(n)\, X_{m-1}(n-1) \ . \tag{A24b}$$

Finally, using a well known matrix inversion lemma, we may relate $\mathcal{R}_{m-1}^{-1}(n-2)$ to $\mathcal{R}_{m-1}^{-1}(n-1)$ as follows:

$$\mathcal{R}_{m-1}^{-1}(n-2) = w \left| \mathcal{R}_{m-1}^{-1}(n-1) + \frac{\mathcal{R}_{m-1}^{-1}(n-1)\, X_{m-1}(n-1)\, X_{m-1}^{T}(n-1) \mathcal{R}_{m-1}^{-1}(n-1)}{1 - \gamma_{m-1}(n-1)} \right| \ . \tag{A25}$$

Substituting Equations A25 and A24b into A24a and combining terms yields Equation A22. Similarly, we may also derive

$$\bar{B}_m(n) = \bar{B}_m(n-1) - \frac{r_m(n,n)}{1 - \gamma_{m-1}(n)} \begin{bmatrix} C_{m-1}(n) \\ \\ 0 \end{bmatrix} \ . \tag{A26}$$

Using Equations A22 and A26, we can relate the error residuals $e_m(n,n)$ and $r_m(n,n)$ to the residuals

$$e_m(n) = e_m(n,n-1) \ , \tag{A27a}$$

and

$$r_m(n) = r_m(n,n-1) \ . \tag{A27b}$$

Specifically, premultipying both sides of Equations A22 and A26 by $X_m^T(n)$,

$$e_m(n,n) = (1 - \gamma_{m-1}(n-1)\,)\, e_m(n) \ , \tag{A28a}$$

and

$$r_m(n,n) = (1 - \gamma_{m-1}(n)\,)\, r_m(n). \tag{A28b}$$

An additional time update relation which may be derived for $k_m(n)$ is

$$k_m(n) = w\, k_m(n-1) + \frac{e_m(n,n)\, r_m(n-1,n-1)}{1 - \gamma_{m-1}(n-1)}$$

$$= w\, k_m(n-1) + (1 - \gamma_{m-1}(n-1)\,)\, e_m(n)\, r_m(n-1) \ . \tag{A29}$$

Equation A29 may be derived using A10 as well as A22 and the time update relation (which follows from Equation 3):

$$V_{m+1}(n) = w\, V_{m+1}(n-1) + x(n-m-1)\, X_m(n) \ . \tag{A30}$$

Lattice recursions can now be derived for $e_m(n)$ and $r_m(n-1)$ by postmultiplying both sides of Equations A9 and A16 (with n replaced by n-1) by $X^T_{m+1}(n)$. The resulting recursions are given by:

$$e_{m+1}(n) = e_m(n) - \frac{k_m(n-1)}{\epsilon^r_m(n-2)}\, r_m(n-1) \ , \tag{A31a}$$

and

$$r_{m+1}(n) = r_m(n-1) - \frac{k_m(n-1)}{\epsilon^e_m(n-1)}\, e_m(n) \ . \tag{A31b}$$

Likewise, an order recursion for $\gamma_m(n)$ may be obtained by premultiplying both sides of Equation A18 by $X^T_m(n)$ and using A28b. The result is

$$\gamma_m(n) = \gamma_{m-1}(n) + \frac{\{ r_m(n)\,(1 - \gamma_{m-1}(n)) \}^2}{\epsilon^r_m(n)} \ . \tag{A32}$$

Equations A15, A17, A29, A31 and A32 represent the least squares adaptive lattice algorithm which generates the residuals $e_m(n)$ and $r_m(n)$. To complete the development of the least squares lattice equalizer algorithm, we return to the problem of obtaining the least squares coefficient vector $F_m(n)$ defined by Equation 2, with N replaced by m. First, note that Equation 2 may be rewritten in the following form:

$$\mathcal{R}_m(n)\, F_m(n) + \bar{Q}_m(n) = 0_{m+1} \ , \tag{A33}$$

where

$$\bar{Q}_m(n) = \sum_{p=0}^{n} w^{n-p}\, a(p)\, X_m(p) \ , \tag{A34}$$

and $0 \leq m \leq N$. It should be pointed out that in replacing N in Equation 2 by m in Equation A33, we are actually considering the larger problem of generating all the least squares coefficient vectors, $F_m(n)$, m=0,1, . . . ,N. As will be seen, in the lattice formulation of the least squares problem, all of these vectors are generated automatically.

In analogy with Equation A5a for the forward prediction vector $\bar{A}_m(n)$, we also have the following augmented matrix equation for $F_m(n)$

$$\bar{\mathcal{R}}_m(n)\, \bar{F}_m(n) = \begin{bmatrix} \rho_m(n) \\ \\ \\ 0_{m+1} \end{bmatrix} \ , \tag{A35}$$

where $\bar{\mathcal{R}}_m(n)$ is an (m+2) x(m+2) dimensional matrix given by:

$$\bar{\mathcal{R}}_m(n) = \sum_{p=0}^{n} w^{n-p}\, \bar{X}_m(p)\, \bar{X}_m^T(p)\ , \tag{A36}$$

and $\bar{X}_m(p)$ is an (m+2) - dimensional column vector given by:

$$\bar{X}_m(p) = (a(p),\, x(p),\, x(p-1),\, \ldots,\, x(p-m)\,)^T. \tag{A37}$$

Also, in Equation A35, the (m+2) - dimensional extended vector, $\bar{F}_m(n)$ is given by:

$$\bar{F}_m(n) = \begin{bmatrix} 1 \\ \\ F_m(n) \end{bmatrix}, \tag{A38}$$

and $\rho_m(n)$ is the minimum of the sum of exponentially weighted least squares error residuals, i.e.,

$$\rho_m(n) = \min \left\{ \sum_{p=0}^{n} w^{n-p}\, (\bar{e}_m(p,n)\,)^2 \right\}$$

$$= \sum_{p=0}^{n} w^{n-p}\, a^2(p) + F_m^T(n)\, \bar{Q}_m(n)\ , \tag{A39}$$

where

$$\bar{e}_m(p,n) = a(p) + F_m^T(n)\, X_m(p). \tag{A40}$$

In analogy with Equations A1, A2 and A3, it easily can be seen from A36 that $\bar{\mathcal{R}}_m(n)$ possesses the following properties:

$$\bar{\mathcal{R}}_m(n) = \left[\begin{array}{c|c} \bar{q}_m(n) & \bar{Q}_m^T(n) \\ \hline \bar{Q}_m(n) & \mathcal{R}_m(n) \end{array}\right] \tag{A41a}$$

$$= \left[\begin{array}{c|c} \bar{\mathcal{R}}_{m-1}(n) & \bar{V}_m(n) \\ \hline \bar{V}_m^T(n) & \bar{v}_m(n) \end{array}\right] \tag{A41b}$$

and

$$\bar{\mathcal{R}}_m(n) = w\, \bar{\mathcal{R}}_m(n-1) + \bar{X}_m(n)\, \bar{X}_m^T(n)\ . \tag{A41c}$$

In Equations A41a and A41b,

$$\bar{q}_m(n) = \sum_{p=0}^{n} w^{n-p}\, a^2(p)\,, \tag{A42a}$$

$$\bar{V}_m(n) = \sum_{p=0}^{n} w^{n-p}\, x(p-m)\, \bar{X}_{m-1}(p)\,, \tag{A42b}$$

and

$$\bar{v}_m(n) = \sum_{p=0}^{n} w^{n-p}\, x^2(p-m) = v_m(n)\,. \tag{A42c}$$

The last equality in Equation A42c follows directly from the definition of $v_m(n)$, Equation A2b.

Using Equations A41, we may now derive order update recursions for $\bar{F}_m(n)$. The key relation which links $F_m(n)$ with the least squares backward prediction residuals is the following recursion:

$$\bar{F}_{m+1}(n) = \begin{bmatrix} \bar{F}_m(n) \\ \\ 0 \end{bmatrix} - \frac{\bar{k}_{m+1}(n)}{\epsilon^r_{m+1}(n)} \begin{bmatrix} 0 \\ \\ \bar{B}_{m+1}(n) \end{bmatrix}, \tag{A43}$$

where the scalar, $\bar{k}_{m+1}(n)$, is given by:

$$\bar{k}_{m+1}(n) = \bar{V}^T_{m+1}(n)\, \bar{F}_m(n)\,. \tag{A44}$$

The derivation of Equations A43 and A44 is analogous to the derivation of A9 and follows by premultiplying the right hand side of A43 by $\bar{R}_{m+1}(n)$ and then using Equations A5b, A35 and A41. Also, we can obtain a time update relation for $\bar{F}_m(n)$. *In particular, we have:*

$$\bar{F}_m(n) = \bar{F}_m(n-1) - \frac{\bar{e}_m(n,n)}{1-\gamma_m(n)} \begin{bmatrix} 0 \\ \\ C_m(n) \end{bmatrix}. \tag{A45}$$

The derivation of Equation A45 is analogous to the derivation of Equation A22 (with $\bar{R}^{-1}_{m-1}(n-2)$ replaced by $\bar{R}^{-1}_m(n-1)$ and $Q_m(n-1)$ replaced by $\bar{Q}_m(n-1)$ *in Equation A24a and* proceeding along a similar development to Equations A24b and A25).

We can now relate the error residual $\bar{e}_m(n,n)$ to the residual:

$$\bar{e}_m(n) = \bar{e}_m(n,n-1)\,. \tag{A46}$$

Specifically, premultiplying both sides of Equation A45 by $\bar{X}^T_m(n)$ yields:

$$\bar{e}_m(n,n) = (1-\gamma_m(n))\, \bar{e}_m(n)\,. \tag{A47}$$

In analogy with Equation A29, the following time update relation for $\bar{k}_m(n)$ may also be derived:

$$\bar{k}_m(n) = w\,\bar{k}_m(n-1) + \frac{\bar{e}_{m-1}(n,n)\,r_m(n,n)}{1-\gamma_{m-1}(n)}$$

$$= w\,\bar{k}_m(n-1) + (1-\gamma_{m-1}(n))\,\bar{e}_{m-1}(n)\,r_m(n). \qquad \text{(A48)}$$

The last equality in Equation A48 follows from A28b and A47. Equation A48 may be derived using A44 as well as A45 and the time update relation, which follows from Equation A41c:

$$\bar{V}_{m+1}(n) = w\,\bar{V}_{m+1}(n-1) + x(n-m-1)\,\bar{X}_m^T(n)\ . \qquad \text{(A49)}$$

Finally, lattice recursion relations for $\bar{e}_m(n)$ may be derived by premultiplying both sides of Equation A43 (with n replaced by n-1 and m+1 replaced by m) by $\bar{X}_m^T(n)$. The resulting recursion is given by:

$$\bar{e}_m(n) = \bar{e}_{m-1}(n) - \frac{\bar{k}_m(n-1)}{\epsilon_m^r(n-1)}\ r_m(n)\ . \qquad \text{(A50)}$$

Equations A50, A48, A15, A17, A29, A31 and A32 make up the complete lattice equalizer algorithm which is summarized in Equations 6, 7 and 8. Note that Equations 7g and 7h are equivalent to A50. It should be pointed out that this algorithm is a modified version of the least squares algorithm originally presented in References 4, 5, 6 and 7. The modification results basically from our use of the residuals $e_m(n)$, $r_m(n)$ and $\bar{e}_m(n)$ as opposed to the residuals $e_m(n,n)$, $r_m(n,n)$ and $\bar{e}_m(n,n)$ which were used in References 4, 5, 6 and 7. We have found that this modified version of the algorithm is more suitable for decision-directed equalization. It should be pointed out that many other modified versions may be readily derived using Equations A28 and A47. As noted in Reference 6, such modifications of the least squares lattice algorithms have proven useful in different applications as they provide trade-offs between computational complexity and storage.

REFERENCES

1. R. W. Lucky, J. Salz, and E. J. Weldon, Jr., Principles of Data Communication, New York: McGraw-Hill, 1968.

2. E. H. Satorius and S. T. Alexander, Channel Equalization using Adaptive Lattice Algorithms, IEEE Transactions on Communications, Vol. Com-27, No. 6, pp 899-905, June 1979.

3. J. D. Markel and A. H. Gray, Jr., Roundoff Noise Characteristics of a Class of Orthogonal Polynomial Structures, IEEE Transactions on Accoustics, Speech and Signal Processing, Vol ASSP-23, pp 473-486, October 1975.

4. M. Morf, D. Lee, J. Nickolls and A. Vieira, Classification of Algorithms for ARMA Models and Ladder Realizations, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Hartford, CT, pp 13-19, May 1977.

5. M. Morf, A. Vieira and D. Lee, Ladder Forms for Identification and Speech Processing, Proceedings of 1977 IEEE Conference on Decision and Control, New Orleans, LA, pp 1074-1078, December 1977

6. M. Morf, Ladder Forms in Estimation and System Identification, 11th Annual Asilomar Conference on Circuits, Systems, and Computers, Monterey, CA, November 7-9, 1977.

7. M. Morf and D. Lee, Recursive Least Squares Ladder Forms for Fast Parameter Tracking, Proceedings of the 1978 IEEE Conference on Decision and Control, San Diego, CA, pp 1362-1367, January 10-12, 1979.

8. D. Godard, Channel Equalization using a Kalman Filter for Fast Data Transmission, IBM Journal of Research and Development, pp 267-273, May 1974.

9. R. D. Gitlin and F. R. Magee, Self-Orthogonalizing Adaptive Equalization Algorithms, IEEE Transactions on Communications, Vol. COM-26, No. 7, July 1977.

10. D. D. Falconer and L. Ljung, Application of Fast Kalman Estimation to Adaptive Equalization, IEEE Transactions on Communications, Vol. COM-26, No. 10, pp 1439-1446, October 1978.

11. J. D. Pack and E. H. Satorius, Least Squares, Adaptive Lattice Algorithms, Naval Ocean Systems Center Technical Report, TR 423, April 1979.

12. E. H. Satorius and M. J. Shensa, On the Application of Recursive Least Squares Methods to Adaptive Processing, presented at the International Workshop on Applications of Adaptive Control, Yale University, New Haven, Connecticut, August 23-25, 1979.

13. M. J. Shensa, Recursive Least Squares Lattice Algorithms – A Geometrical Approach, submitted for publication to IEEE Transactions on Automatic Control.